

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-044576

(43)Date of publication of application : 16.02.1996

(51)Int.Cl. G06F 9/46
G06F 15/00
G06F 15/16

(21)Application number : 07-145885

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 13.06.1995

(72)Inventor : GOSSLER THOMAS
STARK GERHARD

(30)Priority

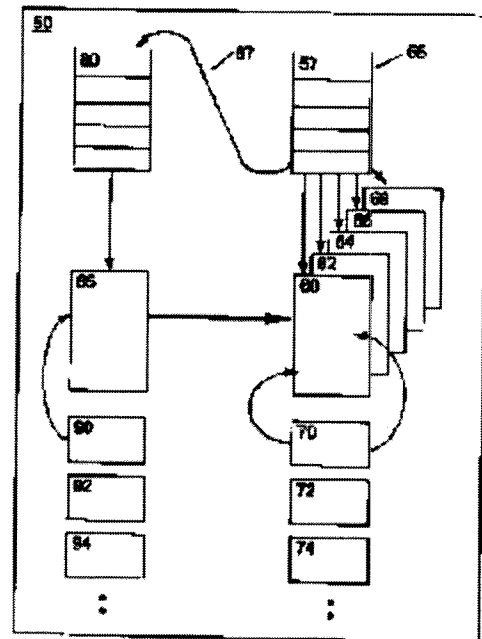
Priority number : 94 94111579 Priority date : 25.07.1994 Priority country : EP

(54) DYNAMIC WORKLOAD BALANCING

(57)Abstract:

PURPOSE: To provide a method for dynamically controlling the number of servers in a transaction system including at least one service unit for processing a service request.

CONSTITUTION: Each service unit 55 includes a queue 57 for receiving an incoming call service request and queuing it, and plural servers 60-68 for executing the service request. This method includes a first step for monitoring the number of the present service requests and the number of the present servers 60-68 assigned to each service unit 55, second step for calculating the optimal number of the servers 60-68 for each service unit 55 according to the number of the present service requests and the number of the present servers 60-68, and third step for assigning the optimal number of servers 60-68 to each service unit 55.



LEGAL STATUS

[Date of request for examination] 10.11.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3610120

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開平8-44576

(43) 公開日 平成8年(1996)2月16日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0 F	7737-5B		
15/00	3 1 0 H	9364-5L		
15/16	3 8 0 Z			

審査請求 未請求 請求項の数 8 O L (全 11 頁)

(21) 出願番号 特願平7-145885

(22) 出願日 平成7年(1995)6月13日

(31) 優先権主張番号 9 4 1 1 1 5 7 9 . 2

(32) 優先日 1994年7月25日

(33) 優先権主張国 ドイツ (DE)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MASCHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72) 発明者 トマス・ゴスラー

ドイツ デー70193 シュトットガルト
シュワーブシュトラッセ 183

(74) 代理人 弁理士 合田 潔 (外2名)

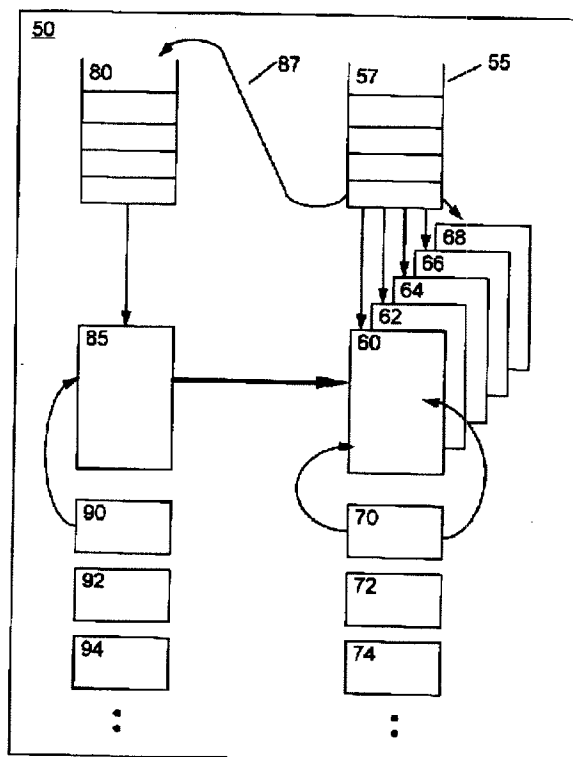
最終頁に続く

(54) 【発明の名称】 動的作業負荷平衡化

(57) 【要約】

【目的】 サービス要求を処理するための少なくとも1つのサービス単位(55)を含むトランザクション・システム内のサーバ(60~68)の数を動的に制御する方法を提供する。

【構成】 各サービス単位(55)は、着信サービス要求を受け取って待ち行列化する待ち行列(57)と、サービス要求を実行する複数のサーバ(60~68)とを含む。本方法は、現在のサービス要求の数と、サービス単位(55)のそれぞれに割り振られている現在のサーバ(60~68)の数とを監視する第1のステップ(240)と、現在のサービス要求の数と、現在のサーバ(60~68)の数とに応じて、各サービス単位(55)ごとにサーバ(60~68)の最適数を求める第2のステップ(260)と、各サービス単位(55)ごとに最適数のサーバ(60~68)を割り振る第3のステップ(260)とを含む。



1

【 特許請求の範囲】

【 請求項1 】 着信サービス要求を受け取って待ち行列化する待ち行列5 7 と、サービス要求を実行する複数のサーバ6 0 ～6 8 とを含み、サービス要求を処理する少なくとも1 つのサービス単位5 5 と、サーバ6 0 ～6 8 を監視し、サーバ6 0 ～6 8 の数を動的に制御する待ち行列化モニタ8 5 と、を含むことを特徴とする、トランザクション・システム。

【 請求項2 】 サービス単位5 5 当たりのサーバの最小数および最大数と、サービス単位5 5 当たりのサーバの最小数および最大数とを組み合わせることにより、サービス単位5 5 内のサービス要求の数に応じて、サービス単位5 5 が使用するサーバ6 0 ～6 8 の数の動的制御を可能にするしきい値とを含むセットアップ・データを待ち行列化モニタ8 5 に提供するセットアップ手段をさらに含むことを特徴とする、請求項1 に記載のトランザクション・システム。

【 請求項3 】 待ち行列化モニタ8 5 を開始する開始手段8 0 をさらに含むことを特徴とする、請求項1 および2 に記載のトランザクション・システム。

【 請求項4 】 サービス要求を処理するための少なくとも1 つのサービス単位5 5 を含むトランザクション・システム内のサーバ6 0 ～6 8 の数を動的に制御する方法において、各サービス単位5 5 が、着信サービス要求を受け取って待ち行列化する待ち行列5 7 と、サービス要求を実行する複数のサーバ6 0 ～6 8 とを含み、前記方法が、現在のサービス要求の数と、サービス単位5 5 のそれぞれに割り振られている現在のサーバ6 0 ～6 8 の数とを監視する第1 のステップ2 4 0 と、現在のサービス要求の数と、現在のサーバ6 0 ～6 8 の数とに応じて、各サービス単位5 5 ごとにサーバ6 0 ～6 8 の最適数を求める第2 のステップ2 6 0 と、各サービス単位5 5 ごとに最適数のサーバ6 0 ～6 8 を割り振る第3 のステップ2 6 0 とを含む方法。

【 請求項5 】 各サービス単位5 5 に割り振られたサーバ6 0 ～6 8 が、永続サーバと一時サーバとを含み、一時サーバのそれぞれが、定義可能な期間または待ち行列5 7 内の少なくとも1 つのサービス要求がそれぞれの一時サーバによって処理可能である間あるいはその両方の間のみ、サービス単位5 5 に割り振られることを特徴とする、請求項4 に記載の方法。

【 請求項6 】 第3 のステップが、最小数のサーバ6 0 ～6 8 をサービス単位5 5 の永続サーバ6 0 ～6 8 として割り振るステップを含むことを特徴とする、請求項4 または5 に記載の方法。

【 請求項7 】 それぞれのサービス単位5 5 内の現在の業務要求の数を定義可能なしきい値で割ることにより、サーバ6 0 ～6 8 の最適数が求められることを特徴とす

2

る、請求項4 ないし6 のいずれかに記載の方法。

【 請求項8 】 サーバ6 0 ～6 8 の数が、それぞれのサービス単位5 5 内のサーバの定義可能な最大数を超えないことを特徴とする、請求項4 ないし7 のいずれかに記載の方法。

【 発明の詳細な説明】

【 0 0 0 1 】

【 産業上の利用分野】 本発明は、メッセージ主導トランザクション環境における作業負荷の動的平衡化に関する。

【 0 0 0 2 】

【 従来の技術】 メッセージ主導トランザクション環境では、ユーザ・インタフェースまたはその他のアプリケーション・プロセスあるいはその両方との通信が、待ち行列に入れられたメッセージに基づいて行われる。メッセージとは、あるクライアント・プロセスからサーバ・プロセスにまたはその逆に送信されるサービス要求または応答(たとえば、資金移動要求、照会要求)であると見なすことができる。

【 0 0 0 3 】 図1 は、典型的なクライアント／サーバ概念を示している。サーバ1 0 は、ネットワーク1 5 内に設置され、少なくとも1 台のクライアント2 0 にサービスを提供する。このようなサービスとしては、たとえば、顧客に関する情報の照会が考えられる。ネットワーク1 5 は、複数のサーバ1 0 と複数のクライアント2 0 とを含むことができ、それにより、各種サービスがクライアントによって要求され、サーバによって提供される。クライアントはサーバからのサービスを要求でき、そのサーバはクライアントとしてさらにサービスを要求できるので、「クライアント」と「サーバ」という用語は置き換えることができる。また、サーバまたはクライアントは、どのようなタイプのプロセス、コンピュータ・プログラムなどでもよいことに留意されたい。サーバ1 0 は、クライアント2 0 からサービス要求を受け取るために要求待ち行列2 5 を含んでいる。サーバ1 0 は、この要求待ち行列2 5 からの要求を処理し、それぞれのクライアント2 0 に適切な応答を送信し、この応答はまずクライアント2 0 の応答待ち行列3 0 に入る。最終的にクライアント2 0 はこの応答待ち行列3 0 から応答を受け取る。並列処理または直列処理あるいはその両方により、クライアントとサーバとの間の複数の対話を同時に処理できることは明らかである。また、それぞれのサーバまたはクライアントが複数の待ち行列を含むことができることも明らかである。

【 0 0 0 4 】 基本的に、クライアント／サーバのシナリオでは、要求の処理のために以下の2 通りの構成のいずれかが必要になる。第1 の構成では、サーバ1 0 が事前始動され、いずれかのクライアントからのサービス要求メッセージが要求待ち行列2 5 に届くのを待つ。要求が要求待ち行列2 5 に届くと、サーバ1 0 は、その要求を

3

処理し、応答を送り返し、次の要求を待つ。

【 0 0 0 5 】 着信要求の並列処理を達成するため、1 台のサーバ1 0 の複数の処理ユニットを事前始動させ、サーバ1 0 の同一要求待ち行列2 5 のサービス要求を処理するためにそのユニットを待機させることもできる。このようにすると、特に、複数のサーバが入出力処理(たとえば、データベースへのアクセス)を実行したり、初期サービス要求に対する応答として他のサーバへの2 次メッセージの流れを開始したりするときに、サーバ1 0 上でのパフォーマンスおよび負荷が改善される。

【 0 0 0 6 】 クライアント / サーバのシナリオの第2 の構成では、サービス要求メッセージが要求待ち行列2 5 に入ったときに、サーバ1 0 だけが(たとえば、開始プロセスによって自動的に)始動される。最終的にその要求が処理されると、サーバ1 0 が終了し、次の要求を処理するために別の処理ユニットが始動される。

【 0 0 0 7 】 パフォーマンスと資源の使用状況を考慮して、適切な機構を選択する必要がある。しかし、どちらの構成でも、システムが動的挙動をするため、条件の厳しい管理方式または複雑な手作業による対話が必要になる。また、1 営業日中でもシステムの作業負荷(たとえば、1 時間当たりのメッセージ数)が変化する(たとえば、ある顧客がテープで多数のサービス要求を提供する場合もあれば、処理する要求がまったくない場合もある)ので、いくつかの問題が発生する可能性がある。

【 0 0 0 8 】 考えられる問題の1 つは、作業負荷が低いために事前始動したサーバ1 0 の処理ユニットが遊んでしまう点である。記憶装置、ディスク空間、またはディスクパッチ・リストなどのシステム資源が占有されるので、他のシステム・ユーザのパフォーマンスに影響する

場合がある。

【 0 0 0 9 】 考えられるもう1 つの問題は、事前始動したサーバ1 0 の処理ユニットの数が少なすぎて、満足できる応答時間内に待ち行列2 5 の多数の要求メッセージを処理できない点である。サーバ1 0 が入出力処理が完了するまで待つから次の要求が処理されるので、このシナリオでは中央演算処理装置(C P U) が十分活用されない可能性がある。サーバ1 0 が使用した処理ユニットが1 台だけの場合には、すべての着信要求が完全に順次処理される。

【 0 0 1 0 】 考えられるもう1 つの問題は、要求メッセージの到着後に追加サーバを始動すると、プロセスの始動と停止のためにシステム・オーバヘッドが高くなる点である。

【 0 0 1 1 】

【 発明が解決しようとする課題】 したがって、本発明の目的は、システム資源のパフォーマンスおよび使用効率がよく、システム・オーバヘッドが低い、ランザクション・システムを提供することにある。

【 0 0 1 2 】

4

【 課題を解決するための手段】 本発明の目的は、着信サービス要求を受け取って待ち行列化する待ち行列5 7

と、サービス要求を実行する複数のサーバ6 0 ~ 6 8 とを含み、サービス要求を処理する少なくとも1 つのサービス単位5 5 と、サーバ6 0 ~ 6 8 を監視し、サーバ6 0 ~ 6 8 の数を動的に制御する待ち行列化モニタ8 5

と、を含むことを特徴とする、ランザクション・システム、およびサービス要求を処理するための少なくとも1 つのサービス単位5 5 を含むランザクション・システム内のサーバ6 0 ~ 6 8 の数を動的に制御する方法

において、各サービス単位5 5 が、着信サービス要求を受け取って待ち行列化する待ち行列5 7 と、サービス要求を実行する複数のサーバ6 0 ~ 6 8 とを含み、前記方法が、現在のサービス要求の数と、サービス単位5 5 のそれぞれに割り振られている現在のサーバ6 0 ~ 6 8 の数とを監視する第1 のステップ2 4 0 と、現在のサービス要求の数と、現在のサーバ6 0 ~ 6 8 の数に応じて、各サービス単位5 5 ごとにサーバ6 0 ~ 6 8 の最適数を求める第2 のステップ2 6 0 と、各サービス単位5 5 ごとに最適数のサーバ6 0 ~ 6 8 を割り振る第3 のステップ2 6 0 とを含む方法によって解決される。

【 0 0 1 3 】 一例として、添付図面に関して本発明について説明する。

【 0 0 1 4 】

【 実施例】 図2 は、本発明によるメッセージ駆動型ランザクション・システムの実施例を示している。このシステムは少なくとも1 つのサービス・ポイント5 0 を含み、さらにこのサービス・ポイントは少なくとも1 つのサービス単位5 5 を含んでいる。明確にするため、本発明の原理を説明するために図2 には1 つのサービス単位5 5 を備えたサービス・ポイントを1 つだけ示す。サービス単位5 5 は、サービス・ポイント5 0 に接続されたクライアント2 0 のいずれかからサービス要求を受け取るサービス単位待ち行列5 7 を含んでいる。このサービス単位待ち行列5 7 は複数のサーバ6 0 ~ 6 8 とリンクされ、それにより、サーバ6 0 ~ 6 8 はサービス単位待ち行列5 7 に待ち行列化されているサービス要求を実行する。サーバ6 0 ~ 6 8 のそれぞれは、要求されたサービスを実行するためにさらに複数のサービス・ルーチン7 0 ~ 7 4 とリンクされており、それにより、サービス・ルーチン7 0 ~ 7 4 のそれぞれが要求されたサービスのモジュラ・ステップを実行する。サーバ6 0 ~ 6 8 のそれぞれは、他のサーバまたは資源管理プログラムにもサービス要求を出すことができることに留意されたい。

【 0 0 1 5 】 図2 では、サーバ6 0 ~ 6 8 の処理の例をサーバ6 0 についてのみ示すが、この例は他のどのサーバについても同様である。サーバ6 0 はサービス単位待ち行列5 7 からサービス要求の1 つを受け取る。サービス要求は、サービスの処理に必要な情報(たとえば、要求されたサービスのタイプ、サービスに必要なデータ)

を提供する。次に、サーバは対応するサービス・ルーチン(この例では、サービス・ルーチン70 および72)をリンクし、サービス・ルーチン70 および72を使用することにより要求されたサービスを実行する。

【0016】サービス・ポイント50は、待ち行列化モニタ85に接続された開始待ち行列80をさらに含んでいる。サービス単位待ち行列57に最初のサービス要求が入ると、待ち行列化システムによってトリガ・メッセージ87が生成され、それが開始待ち行列80に送られる。待ち行列化モニタ85は、サービス単位待ち行列57の名前と始動するサーバの名前とが入っているトリガ・メッセージ87を受け取る。次に、待ち行列化モニタ85は、サービス・ポイント50の各サービス単位55についてサーバ60～68を監視し制御する。待ち行列化モニタ85が使用する監視および制御の方法については、後述する。

【0017】図3は、複数のサービス・ポイント50、100、および102を有するメッセージ主導トランザクション・システムの実施例を示しているが、これらのサービス・ポイントはさらに複数のサービス単位55、104、および120を含んでいる。明確にするため、図2と同じ特徴を示す参照番号はそのまま使用する。たとえば、サービス・ポイント50は、複数のサービス単位55、104、および120を示している。サービス単位104は、サーバ110および112にリンクされたサービス単位待ち行列105を含んでいる。また、サービス単位120は、サーバ130～136にリンクされたサービス単位待ち行列125を含んでいる。サーバ60～64、110～112、および130～136のそれぞれは複数のサービス・ルーチン150～174にリンク可能であり(図3にはそれらのリンクの一部のみ示す)、それぞれのサービス要求内の情報に応じたサービスを提供する。また、それぞれのサーバは、初期サービス要求を処理するために2次要求になりうるサービス要求を出すこともできる。

【0018】サービス単位55、104、および120のそれぞれは、サービス単位待ち行列57、105、または125がそれぞれの最初のサービス要求を受け取ったときに、それぞれのトリガ・メッセージを開始待ち行列80に送信する。待ち行列化モニタ85は、そのトリガ・メッセージを受け取り、各サービス単位55、104、および120ごとに複数のセットアップ・プロファイル90～94からの対応するセットアップ・プロファイルをリンクする。各セットアップ・プロファイル90～94は複数のサービス単位についてリンク可能で、特定のサービス単位1つについてののみカスタマイズされるのではないことは明らかである。次に、待ち行列化モニタ85は、各サービス単位55、104、および120ごとにサーバの数を監視し制御する。

【0019】それに応じてサービス・ポイント100お

よび102を構築することができ、それにより、サービス単位の数と各サービス・ポイントでの実行回数が増える可能性がある。

【0020】図2および図3の待ち行列化モニタ85は、サーバ10の各種プロセスに対して動的作業負荷平衡化と定義済みの構造(図7で説明する)を提供する。動的作業負荷平衡化については以下に説明するが、その際、明確にするために図2の参照番号を使用することが好ましい。

【0021】待ち行列化モニタ85は、以下の顧客定義のサービス単位パラメータを含む指定のセットアップ・プロファイル90～94から、監視対象の各サービス単位55に関する制御情報を受け取る。

1. 監視対象のサービス単位待ち行列57(複数可)の名前。これにより、1つの待ち行列化モニタ85によって1つのサービス・ポイント50内の複数のサービス単位待ち行列を監視することができる。
2. 着信サービス要求の処理のためのそれぞれのサービス単位55の関連サーバ(複数可)60～68の名前。
3. 各サービス単位55のサーバの最小数。この最小数は、処理すべきメッセージがない場合でも永続的に動作する必要があるサービス単位55内のサーバ・プロセスの数に相当する。この最小数の値は、各サービス単位待ち行列57ごとに指定することができる。
4. 各サービス単位55のサーバの最大数。この最大数は、待ち行列に多くの業務要求が入っている高作業負荷状況で動作する必要があるサービス単位55内のサーバ・プロセスの数に相当する。この最大数の値は、各サービス単位待ち行列57ごとに指定することができる。
5. それぞれのサービス単位待ち行列内の業務要求を迅速に処理するためにリンクする必要があるサーバの数を定義するしきい値。リンクする必要があるサーバの数は、待ち行列内の業務要求の数(待ち行列内項目数)をしきい値で割った数によって決まる。サーバの最大数と、リンクする必要があるサーバの数とを組み合わせると、最終的に、各サービス単位55によってリンクされるサーバ60～68の数が決まる。

6. 待ち行列化モニタ85がサービス単位(複数可)55の現在の状態の監視を繰り返す時間間隔を定義するモニタ時間間隔。

7. サービス単位55をただちに始動するか、トリガ・メッセージ87の到着後に始動するかを決定する自動始動指示。

【0022】作業負荷状況の変化に対応するため、上記のパラメータはいずれもユーザが動的に変更することができる。

【0023】待ち行列化モニタ85は永続的に動作し、実行時環境が開始されたときに自動的に始動する。待ち行列化モニタ85は、サービス・ポイント50内で待ち

行列化モニタ85が監視する各サービス単位55、104、および120ごとに最適数のサーバを使用する。図4は、監視する各サービス単位ごとに最適数のサーバを使用するために待ち行列化モニタ85または他の待ち行列化モニタが提供する動的作業負荷平衡化方法を示している。

【0024】ステップ200で、待ち行列化モニタ85は、トリガ・メッセージ(複数可)87が開始待ち行列80に到着するのを待ち、次にステップ210で、リンクされたセットアップ・プロファイルの定義通り、各サービス単位55、104、および120ごとに指定の最小数のサーバを始動する。ただし、220の自動始動指示に応じて、待ち行列化モニタの始動後ただちに待ち行列化モニタ85によってステップ210の最小数のサーバを始動することもできる。

【0025】待ち行列化モニタ85は、ステップ210に続く次のステップ230で、リンクするサーバのパラメータとして、処理する必要があるサービス単位待ち行列の名前と、状態指示とを提供する。この状態指示は、それぞれのサーバが永続的に使用されるのか、一時的に使用されるだけなのかを決定するものである。永続サーバはシステムが終了するまでそれぞれのサービス単位にリンクされるが、一時実行マネージャは、サービス単位待ち行列からの1つのサービス要求を処理する場合のみ、それぞれのサービス単位にリンクされる。このサービス要求の処理が完了したときに処理すべきサービス要求が他になければ、この一時サーバへのリンクが終了するか、サーバ自体が終了する。

【0026】最小数のサーバとしてリンクされているサーバは永続サーバとして示されるので、サービス単位当たりの永続サーバの数は、通常の条件下でサービス単位パラメータによって示されるサーバの指定の最小数によって決まる。しかし、サービス単位当たりの一時サーバの数は、指定のサービス単位パラメータと、それぞれのサービス単位待ち行列57内のサービス要求の数とによって決まる。

【0027】指定のモニタ時間間隔が経過すると、待ち行列化モニタ85は、次のステップ240でサービス単位55(104および120)のそれぞれから以下の情報を問い合わせる。

1. それぞれのサービス単位待ち行列内の現在のサービス要求の数。これは待ち行列内項目数ともいう。
2. このサービス単位待ち行列にリンクされている現在のサーバの数。

【0028】それぞれのサービス単位55内の現在のサービス要求(SR)の数が指定のしきい値以下であるときは、待ち行列化モニタ85は追加のサーバを始動する必要はない。保全性を維持するため、待ち行列化モニタ85は、ステップ250で最小数のサーバが動作していることを検査する。動作しているサーバが指定の最小数

のサーバより少ない場合は、待ち行列化モニタ85は、このサーバの最小数に達するために必要な数のサーバのみ再始動する。

【0029】サービス単位待ち行列57内のサービス要求の数が指定のしきい値を上回る場合は、待ち行列化モニタ85は、以下の式に基づいてステップ260で追加のサーバをリンクまたは始動する。

【0030】

【数1】リンク／始動するサーバの数＝待ち行列内項目数／しきい値

【0031】リンク／始動するサーバの数の計算値にすでに動作しているサーバの数を加えた合計がサーバ数の指定の最大値を上回る場合は、待ち行列化モニタ85は、このサーバの最大数に達するために必要な数のサーバのみリンクまたは始動する。

【0032】ステップ250または260のいずれかに続いて、もう一度ステップ230に戻り、そこでステップ250で再始動されたサーバが永続サーバとして示され、ステップ260で始動された追加のサーバが一時サーバとして示される。

【0033】実行時環境が終了すると、待ち行列化モニタは、監視したすべてのサービス単位待ち行列を使用禁止にし、サーバに対してこの事象に関する通知を行う。

【0034】図5および図6は、動的作業負荷平衡化の例を示している。図5の例では、待ち行列化モニタ85は、指定のセットアップ・プロファイル90～94から以下の顧客定義のサービス単位パラメータを受け取る。

1. 監視するサービス単位待ち行列57
3. サーバの最小数は3である。
4. サーバの最大数は10である。
5. しきい値は2である。

【0035】次に、待ち行列化モニタ85は、サービス単位待ち行列57の現在の待ち行列内項目数を要求し、この例では「待ち行列内項目数は6である」というメッセージを受け取る。さらに待ち行列化モニタ85は、サービス単位55で使用されている現在のサーバの数を要求し、「3台の永続サーバと2台の一時サーバが使用され、現在のサーバの数は5である」というメッセージを受け取る。

【0036】ここで待ち行列化モニタ85は、リンクするサーバの数を決定する。待ち行列内項目数(=6)をしきい値(=2)で割ると、リンクする必要がある一時サーバの数が3になる。ただし、待ち行列化モニタ85は、3台の一時サーバをリンクする前に、使用するサーバの総数がサーバの最大数(=10)を超えていないかどうか検査する。すでに使用している5台のサーバに、使用する必要がある3台のサーバを加えると、合計8台のサーバになるはずだが、10台のサーバという所与の最大数を超えない。これは、サービス単位待ち行列57からのサービス要求を迅速に処理するために待ち行列化

モニタが最終的に3 台のサーバを一時サーバとしてリンクすることを意味する。

【 0 0 3 7 】 図6 の例では、待ち行列化モニタ8 5 は、指定のセットアップ・プロファイル9 0 ~ 9 4 から図5 の例と同じサービス単位パラメータを受け取るが、以下の点のみ異なっている。

3 . サーバの最小数は4 である。

5 . しきい値は3 である。

【 0 0 3 8 】 次に、待ち行列化モニタ8 5 は、サービス単位待ち行列5 7 の現在の待ち行列内項目数とサービス単位5 5 で使用されている現在のサーバの数とをもう一度要求し、「待ち行列内項目数は9 であり、4 台の永続サーバと4 台の一時サーバが使用され、現在のサーバの数は8 である」というメッセージを受け取る。

【 0 0 3 9 】 待ち行列内項目数(= 9) をしきい値(= 3) で割ると、リンクする必要がある一時サーバの数が3 になる。ただし、すでに使用されている8 台のサーバに追加して使用する必要があるサーバ3 台を加えると合計1 1 台のサーバになり、1 0 台のサーバという所与の最大数を1 台上回るので、使用するサーバの総数がサーバの最大数(= 1 0) を超えるはずである。これは、待ち行列化モニタがサービス単位待ち行列5 7 に対して2 台の追加サーバしか一時サーバとしてリンクできないことを意味する。

【 0 0 4 0 】 図7 は、サーバによるサービス要求の処理を示すものである。ステップ3 0 0 で、サーバ6 0 ~ 6 8 は、ステップ2 3 0 で待ち行列化モニタ8 5 によって提供されたパラメータを受け取る。このパラメータは、処理する必要があるサービス単位待ち行列の名前と、それぞれのサーバが永続的に使用されるのか、一時的のみ使用されるのかを示す状態指示とを含んでいる。

【 0 0 4 1 】 次のステップ3 1 0 で、サーバ6 0 ~ 6 8 は、それぞれのサービス単位待ち行列5 7 によってサービス単位が割り振られるように所定の待ち時間待機する。サーバの待ち時間の値は、状態指示によって決まる。一時サーバの待ち時間は定義可能な期間であるが、永続サーバの待ち時間は不定として設定される。すなわち、一時サーバは、ゼロにもなりうる定義可能な期間だけ待機するが、永続サーバは、サービス単位待ち行列からサービス要求が最終的に割り振られるまで待機することになる。

【 0 0 4 2 】 指定の待ち時間内にサービス単位待ち行列5 7 にサービス要求が入っている場合、そのサービス要求はステップ3 2 0 でそれぞれのサーバによって読み取られ、処理される。サービス要求が最終的に処理されると、サーバは、処理したサービス要求の送信側(クライアント) に応答信号を出し、ステップ3 1 0 に戻る。

【 0 0 4 3 】 指定の待ち時間内にサービス単位待ち行列5 7 にサービス要求が一切入っていない場合(たとえば、待ち行列が空であるか、使用禁止になっていると

き) は、一時サーバが終了し、そのサービス単位へのリンクはステップ3 3 0 で解除される。

【 0 0 4 4 】 前述の待ち行列化モニタとサーバ構造により、メッセージ・システム内で最良の資源使用効率と応答時間を達成することが可能になる。この平衡化方法は完全に動的なものなので、システムの動作中にすべての構成データを変更することができる。

【 0 0 4 5 】 リンクするサーバの数を決定するために提供される方法は前述の実施例に限定されないことに留意されたい。サービス単位内の現在のサーバの数を現在の作業負荷に動的に適応させるという要件を満たすものであれば、他の方法も適用可能である。

【 0 0 4 6 】 オブジェクト指向アプリケーションでは、本明細書で使用する「サーバ」という用語が「サーバ・インスタンス」("Object-oriented Software Construction", Bertrand Meyer 著、1988年、Prentice Hall International Ltd(英国) 発行、ISBN 0-13-629049-3、71 ページ、5.2.1章を参照) という用語で置換え可能であることに留意されたい。

【 0 0 4 7 】 まとめとして、本発明の構成に関して以下の事項を開示する。

【 0 0 4 8 】 (1) 着信サービス要求を受け取って待ち行列化する待ち行列5 7 と、サービス要求を実行する複数のサーバ6 0 ~ 6 8 とを含み、サービス要求を処理する少なくとも1 つのサービス単位5 5 と、サーバ6 0 ~ 6 8 を監視し、サーバ6 0 ~ 6 8 の数を動的に制御する待ち行列化モニタ8 5 と、を含むことを特徴とする、トランザクション・システム。

(2) サービス単位5 5 当たりのサーバの最小数および最大数と、サービス単位5 5 当たりのサーバの最小数および最大数とを組み合わせることにより、サービス単位5 5 内のサービス要求の数に応じて、サービス単位5 5 が使用するサーバ6 0 ~ 6 8 の数の動的制御を可能にするしきい値とを含むセットアップ・データを待ち行列化モニタ8 5 に提供するセットアップ手段をさらに含むことを特徴とする、上記(1) に記載のトランザクション・システム。

(3) 待ち行列化モニタ8 5 を開始する開始手段8 0 をさらに含むことを特徴とする、上記(1) および(2) に記載のトランザクション・システム。

(4) サービス要求を処理するための少なくとも1 つのサービス単位5 5 を含むトランザクション・システム内のサーバ6 0 ~ 6 8 の数を動的に制御する方法において、各サービス単位5 5 が、着信サービス要求を受け取って待ち行列化する待ち行列5 7 と、サービス要求を実行する複数のサーバ6 0 ~ 6 8 とを含み、前記方法が、現在のサービス要求の数と、サービス単位5 5 のそれぞれに割り振られている現在のサーバ6 0 ~ 6 8 の数とを監視する第1 のステップ2 4 0 と、現在のサービス要求の数と、現在のサーバ6 0 ~ 6 8 の数とに応じて、各サ

サービス単位55ごとにサーバ60～68の最適数を求める第2のステップ260と、各サービス単位55ごとに最適数のサーバ60～68を割り振る第3のステップ260とを含む方法。

(5) 各サービス単位55に割り振られたサーバ60～68が、永続サーバと一時サーバとを含み、一時サーバのそれぞれが、定義可能な期間または待ち行列57内の少なくとも1つのサービス要求がそれぞれの一時サーバによって処理可能である間あるいはその両方の間のみ、サービス単位55に割り振られることを特徴とする、上記(4)に記載の方法。

(6) 第3のステップが、最小数のサーバ60～68をサービス単位55の永続サーバ60～68として割り振るステップを含むことを特徴とする、上記(4)または(5)に記載の方法。

(7) それぞれのサービス単位55内の現在の業務要求の数を定義可能なしきい値で割ることにより、サーバ60～68の最適数が求められることを特徴とする、上記(4)ないし(6)のいずれかに記載の方法。

(8) サーバ60～68の数が、それぞれのサービス単位55内のサーバの定義可能な最大数を超えないことを特徴とする、上記(4)ないし(7)のいずれかに記載の方法。

【0049】

【発明の効果】本願発明によれば、サービス要求を処理するための少なくとも1つのサービス単位(55)を含むトランザクション・システム内のサーバ(60～68)の数を動的に制御することができることとなった。

【図面の簡単な説明】

【図1】典型的なクライアント／サーバの概念を示す図

である。

【図2】本発明によるメッセージ主導トランザクション・システムの実施例を示す図である。

【図3】複数のサービス・ポイントを有するメッセージ主導トランザクション・システムの実施例を示す図である。

【図4】待ち行列化モニタによって提供される動的作業負荷平衡化方法を示す図である。

【図5】動的作業負荷平衡化の例を示す図である。

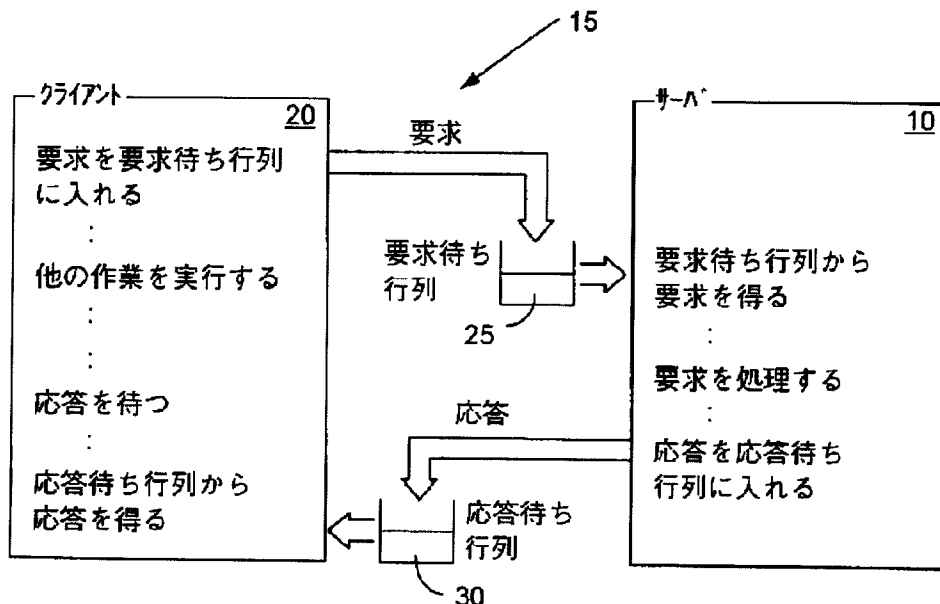
【図6】動的作業負荷平衡化の例を示す図である。

【図7】サーバによるサービス要求の処理を示す図である。

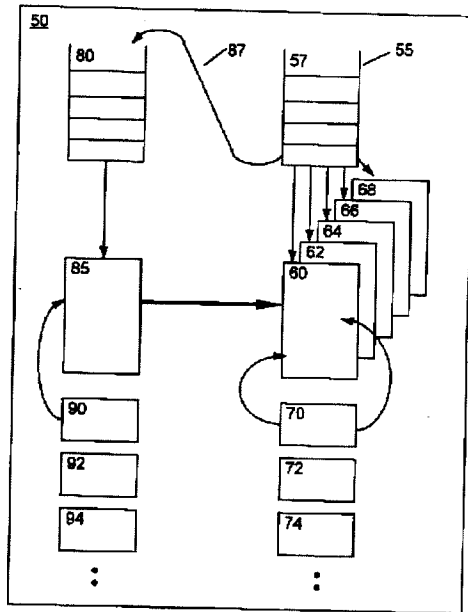
【符号の説明】

50	サービス・ポイント
55	サービス単位
57	サービス単位待ち行列
60	サーバ
62	サーバ
64	サーバ
66	サーバ
68	サーバ
70	サービス・ルーチン
72	サービス・ルーチン
74	サービス・ルーチン
80	開始待ち行列
85	待ち行列化モニタ
87	トリガ・メッセージ
90	セットアップ・プロファイル
92	セットアップ・プロファイル
94	セットアップ・プロファイル

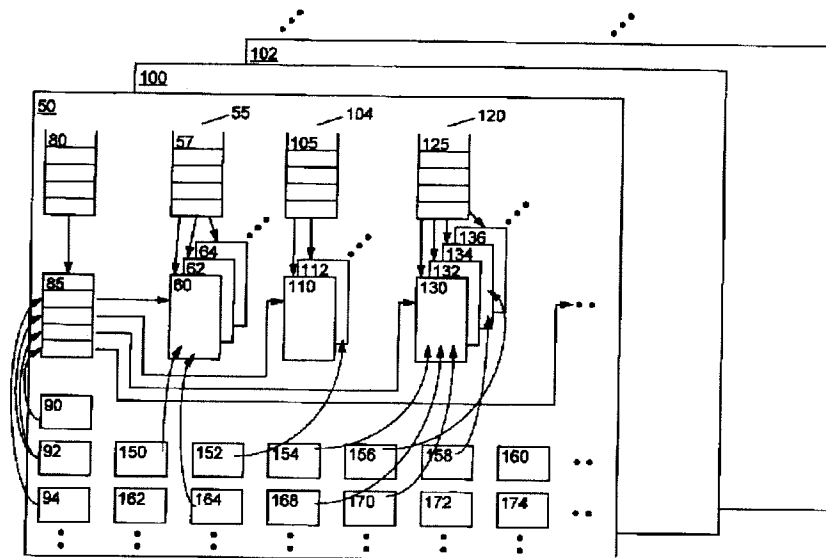
【図1】



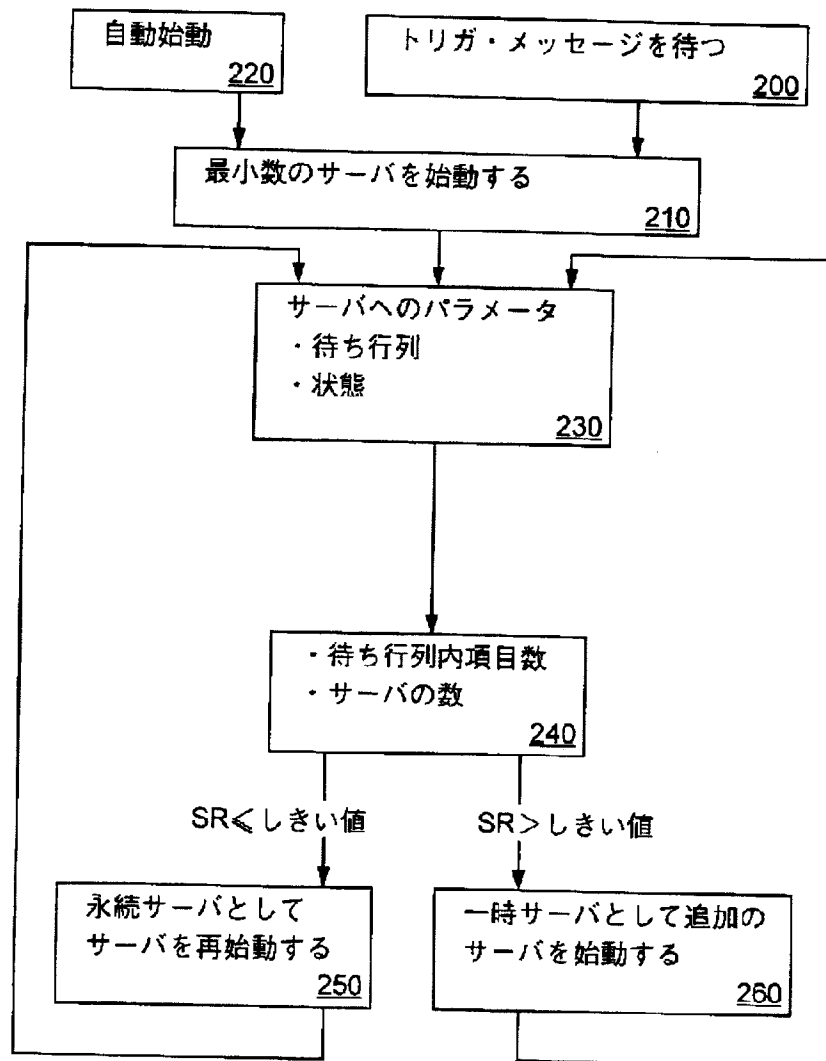
【 図2 】



【 図3 】



【 図4 】



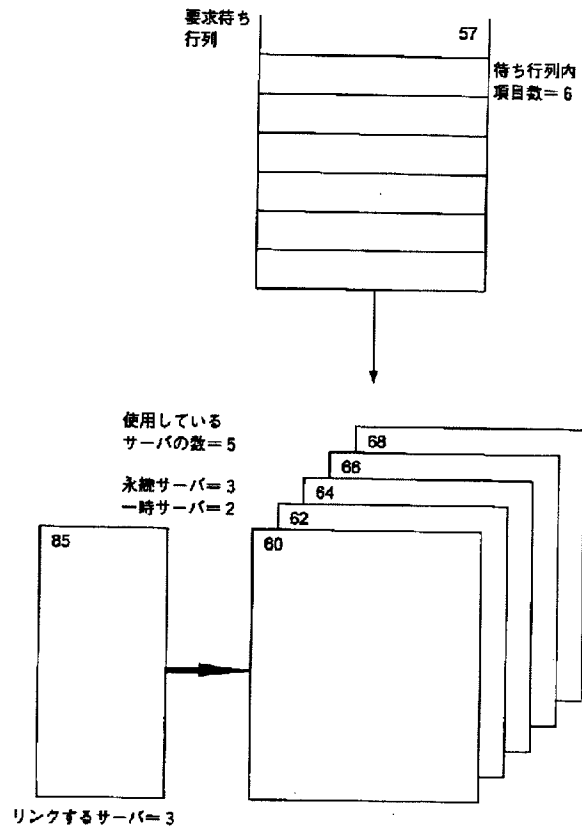
【 図5 】

構成済みパラメータ：

しきい値＝2

サーバの最小数＝3

サーバの最大数＝10



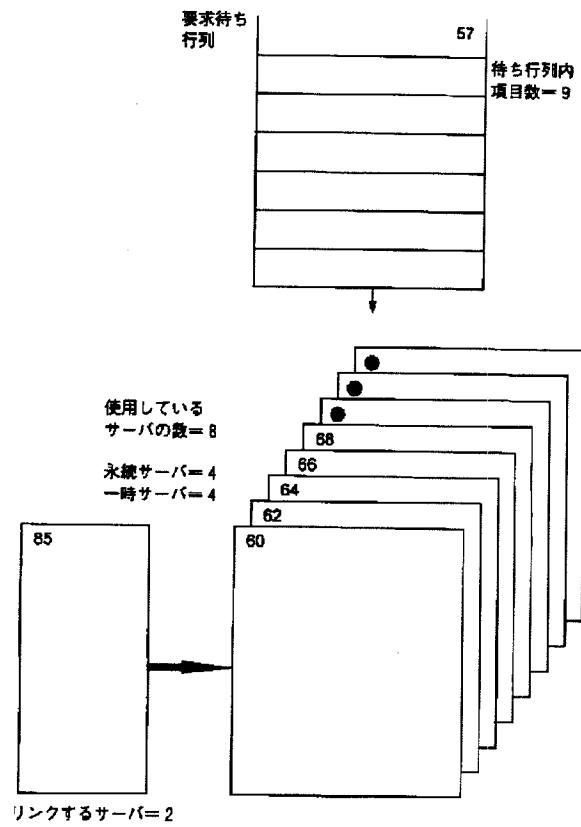
【 図6 】

構成済みパラメータ：

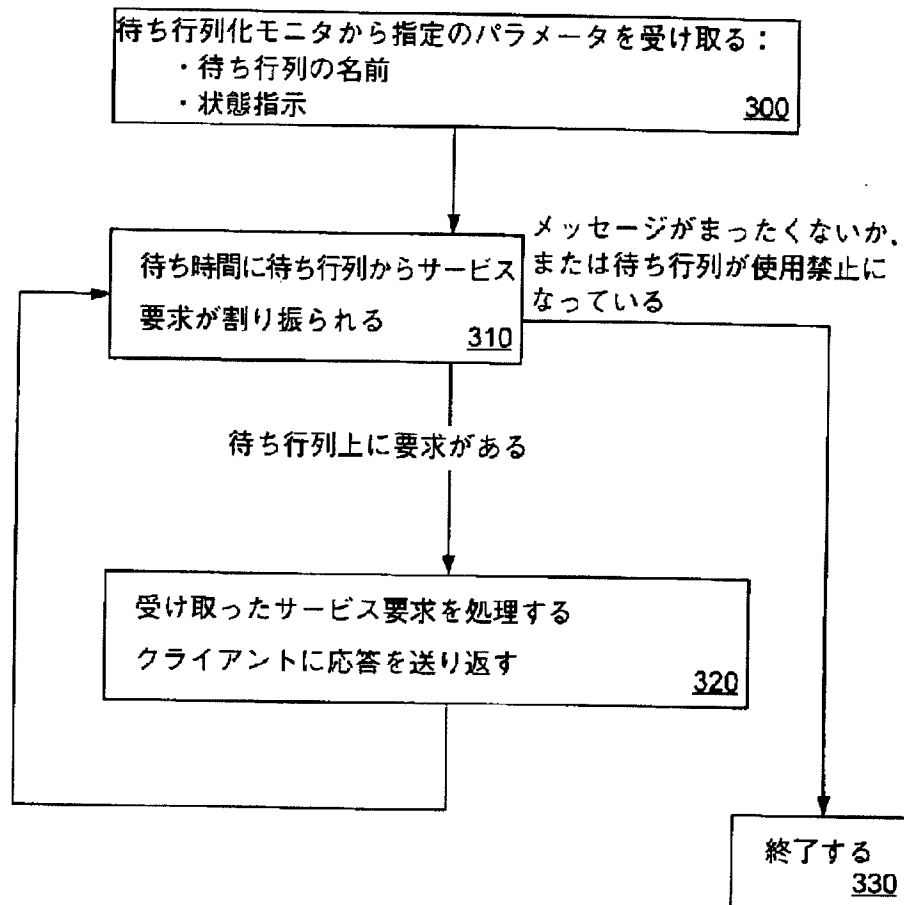
しきい値＝3

サーバの最小数＝4

サーバの最大数＝10



【 図7 】



フロント ページの続き

(72)発明者 ゲルハルト・シュタルク
ドイツ ディー75382 アルト ヘングシュ
テット トウルペンシュトラッセ 25